

微机原理

硬件实验报告

实验一 I/O 地址译码

一、实验目的

- 1、掌握 I/O 地址译码电路的工作原理。

二、实验内容及原理

实验电路如图1-1所示，其中74LS74为D触发器，可直接使用实验台上数字电路实验区的D触发器，74LS138为地址译码器。译码输出端Y0~Y7在实验台上“I/O地址”输出端引出，每个输出端包含8个地址，Y0: 280H~287H, Y1: 288H~

28FH, …… 当CPU执行I/O指令且地址在280H~2BFH范围内, 译码器选中, 必有一根译码线输出负脉冲。

根据图1-1, 我们可以确定A9~A3, AEN, IOW, IOR的值。要使译码电路正常工作, 必须使处于低电平有效。因而可以确定A6=A8=0, A7=A9=1, AEN=0, IOW与IOR不可同时为1 (即不能同时读写)。当要从Y4输出低脉冲时, A5A4A3=100; 从Y5输出时, A5A4A3=101。综上所述, Y4输出时, 应设置值2A0H (A9~A0=1010100000B); Y5输出时, 应设置值2A8H (A9~A0=1010101000B)。

执行下面两条指令

MOV DX, 2A0H

OUT DX, AL (或IN AL, DX)

Y4输出一个负脉冲到D触发器的CLK上, 因为D=1 (接了高电平+5V), 所以Q被赋值为1。

延时一段时间(delay);

执行下面两条指令;

MOV DX, 2A8H

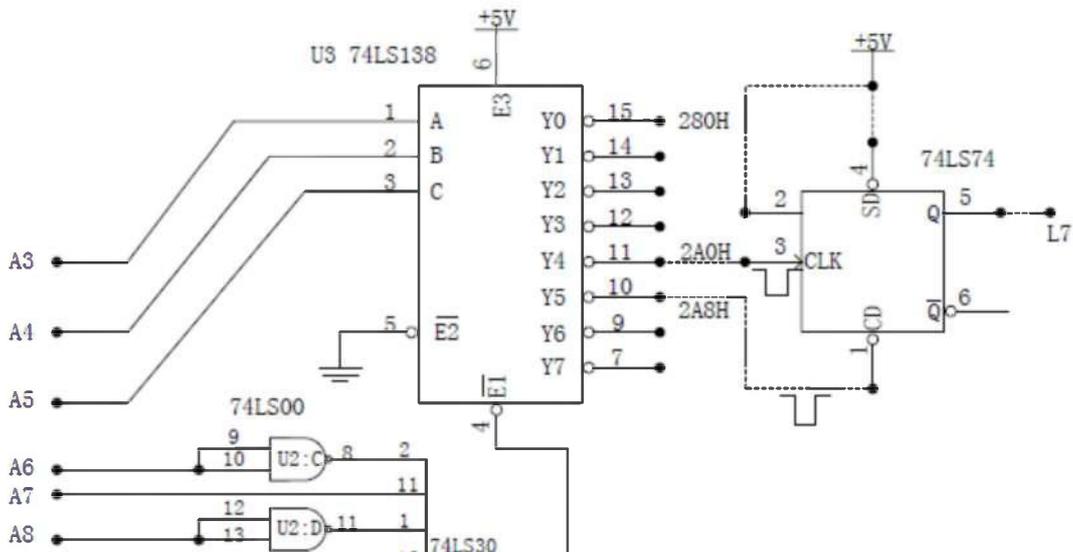
OUT DX, AL (或IN AL, DX)

Y5输出一个负脉冲到CD, D触发器被复位, Q=0。

再延时一段时间, 然后循环上述步骤。

利用这两个个负脉冲控制L7闪烁发光 (亮、灭、亮、灭、……), 时间间隔通过软件延时实现。

三、硬件接线图与软件流程图



硬件接线：

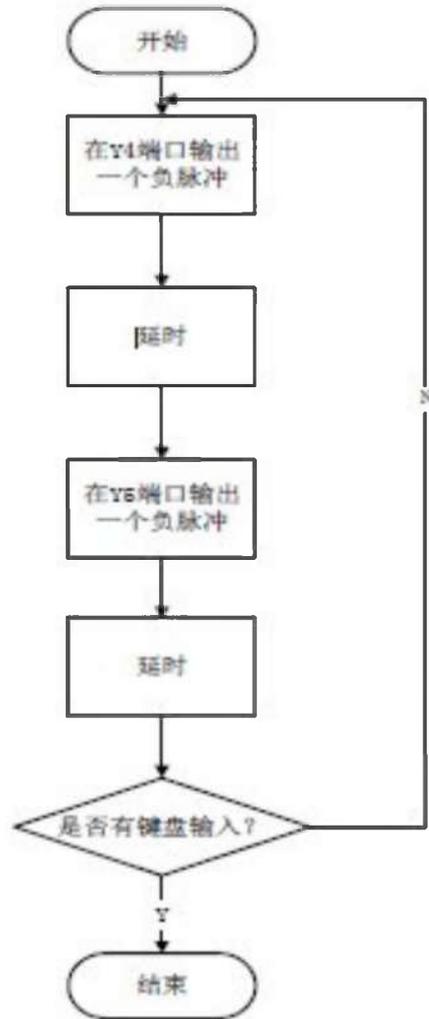
Y4/IO 地址 接 CLK/D 触发器

Y5/IO地址 接 C/D触发器

D/D触发器 接 SD/D角发器 接+5V

Q/D触发器 接 L7（LED灯）或逻辑笔

软件流程图：



四、源程序

OUTPORT1 EQU 2A0H ;预置，方便修改

OUTPORT2 EQU 2A8H

CODE SEGMENT

ASSUME CS:CODE

START: MOV DX,OUTPORT1 ;根据原理图设定A9~A0 的值 (Y4)

OUT DX,AL ;让译码器Y4 口输出一个负脉冲

CALL DELAY ;延时

MOV DX,OUTPORT2 ;根据原理图设定A9~A0 的值 (Y5)

OUT DX,AL ;让译码器Y5 口输出一个负脉冲

CALL DELAY ;延时

MOV AH,1 ;调用1 号DOS 功能，等待键盘输入

```
INT 16H
JE START ;若有键盘输入则退出程序，否继续循环
MOV AH,4CH
INT 21H
DELAY PROC NEAR ;延时子程序
MOV BX,200 ;时延长度 (200)
A: MOV CX,0
B: LOOP B
DEC BX
JN E A
RET
DELAY ENDP
CODE ENDS
END START
```

五、实验结果

LED 灯处于闪烁状态，键盘有输入后，成功退出。

六、实验总结

遇到的问题：需要合理设置时延值。

七、实验收获与心得体会

这是第一次进行微机原理的硬件实验，熟悉了使用的实验系统，学习了最基本的硬件测试方法，学会了在实验的环境下进行单步调试，对接口的地址、LED灯的控制有了初步的了解。为后续实验做了准备。

实验二 简单并行接口

一、实验目的

- 1、掌握简单并行接口的工作原理及使用方法。

二、实验内容及原理

1、按下面图 2-1 简单并行输出接口电路图连接线路(74LS273 插通用插座, 74LS32 用实验台上的“或门”)。74LS273 为八 D 触发器, 8 个 D 输入端分别接数据总线 D0~D7, 8 个 Q 输出端接 LED 显示电路 L0~L7。

2、编程从键盘输入一个字符或数字, 将其 ASC II 码通过这个输出接口输出, 根据 8 个发光二极管发光情况验证正确性。

三、硬件接线图与软件流程图

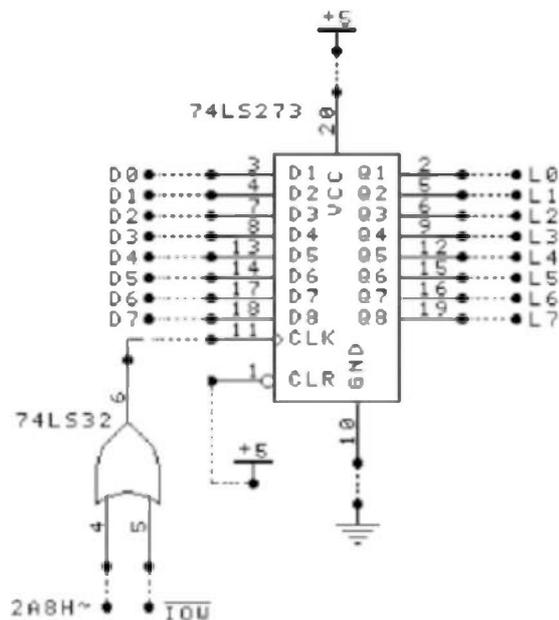
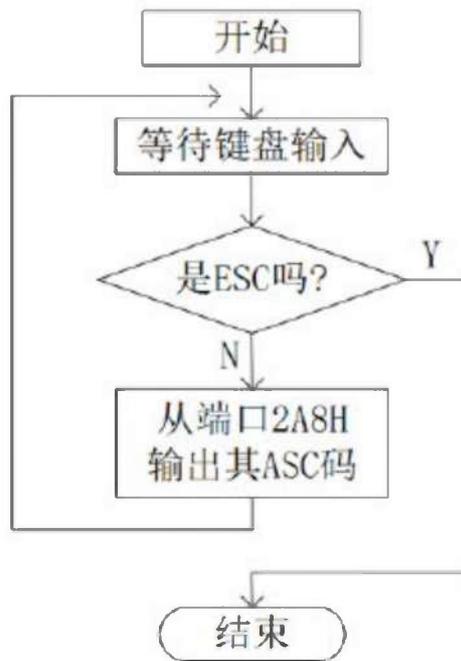


图2-1

软件流程图:



四、源程序

```

CODE SEGMENT
    ASSUME CS:CODE
START: MOV CX,0FFH
LOOP1: MOV AH,1
        INT 21H
        CMP AL,1BH
        JZ END1
        MOV DX,2A8H
        OUT DX,AL
        LOOP LOOP1
END1: MOV AX,4C00H
        INT 21H
CODE ENDS
    END START
  
```

五、 实验结果

键盘输入一个字符，LED 灯显示相应的 ASCII 码。键入 ESC 后成功退出。

六、 实验收获与心得体会

本次实验让我对硬件实验有了更加深刻的理解，硬件实验需要经常读取硬件的状态，根据其状态采取相应的控制方案，同时需要向接口输出数据和控制信息，驱动硬件正常工作。通过键盘的输入还记下了一些字符的 ASCII 码。

实验三 可编程并行接口8255

一、 实验目的

1、通过实验，掌握8255工作于方式0以及设置A口为输出口，C口为输入口的的方法。

二、 实验内容及原理

1、 8255芯片工作原理：

1) 8255的工作方式：

一片8255内部有3个端口，A口可以工作在方式0、方式1或方式2，B口可以工作在方式0、方式1，C口可以工作在方式0。

方式0是基本型输入/输出。这种方式和外设交换数据时，8255端口与外设之间不使用联络线。

方式1为选通型输入/输出。用这种方式和外界交换数据时，端口和外设之间要有联络信号。

方式2是双向数据传送，仅A口有这项功能。当A口工作在方式2时，B口仍可以工作在方式0或方式1，但此时B口方式1只能用查询方式与CPU交换信息。

2) 工作方式选择字

8255工作方式选择字共8位（如图3-1），存放在8255控制寄存器中。最高位D7为标志位，D7=1表示控制寄存器中存放的是工作方式选择字，D7=0表示控制寄存器中存放的是C口置位/复位控制字。

	D7	D6 D5	D4	D3	D2	D1	D0
1		00-方式0	A口	C口高四位	0-方式0	B口	C口低四位
		01-方式1	1: 输入	1: 入	1-方式1	1: 入	1: 入
		1x-方式2	0: 输出	0: 出		0: 出	0: 出

标志位
A组
B组

根据上图，方式控制字应为：10000000B，即80H。

8255控制寄存器端口地址--28BH, A口的地址--288H, B口的地址--289H, C口的地址--28AH。

3) C口置/复位控制字

8255的C口可进行位操作，即：对C口可进行位操作，即：对8255C口的每一位进行置或清零操作，该操作通过设置C口置/复位字实现的（图）。复位字实现的（图）。C口置/复位字共8位，各位含义如下：

D7	D6 D5 D4	D3 D2 D1	D0
0		000 : PC0 001 : PC1 111 : PC7	1 : 置位 0 : 复位
标志位	x x x		

2、8255A的控制信号与传输动作的对应关系

A1	A0	/RD	/WR	/CS	工作状态
0	0	0	1	0	A口数据→数据总线
0	1	0	1	0	B口数据→数据总线
1	0	0	1	0	C口数据→数据总线
0	0	1	0	0	数据总线→A口数据
0	1	1	0	0	数据总线→B口数据
1	0	1	0	0	数据总线→C口数据
1	1	1	0	0	数据总线→控制寄存器
X	X	X	X	1	数据总线→三态
1	1	0	1	0	非法状态
X	X	1	1	0	数据总线→三态

3、命令字与初始化编程

8255有两个命令字，即方式选择控制字和C口置0/置1控制字，初始化编程的步骤是：

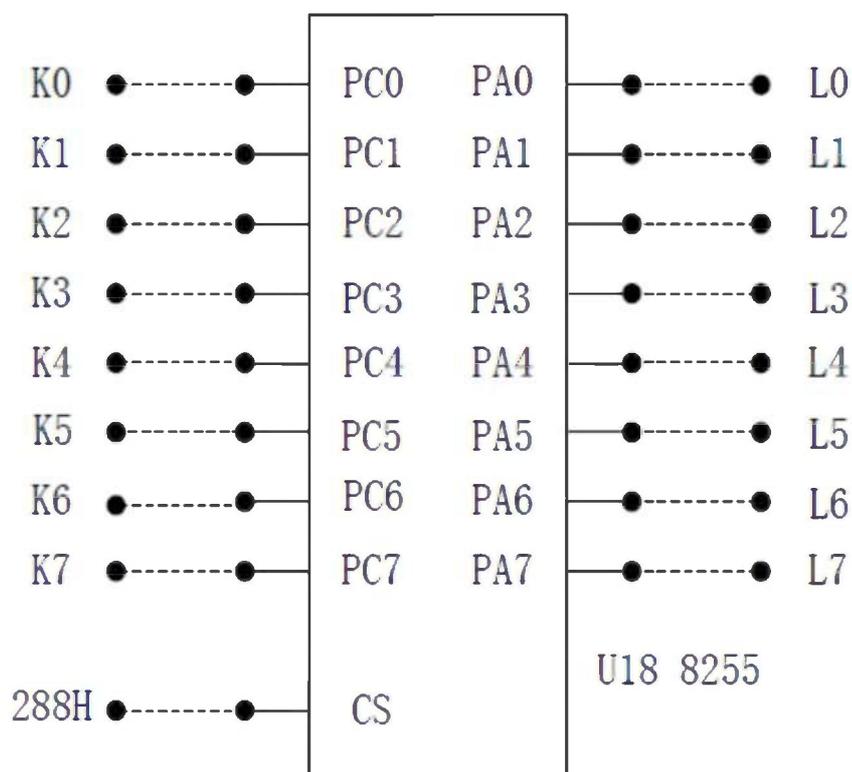
- ① 向8255控制寄存器写入“方式选择控制字”，从而预置端口的工作方式；
- ② 当端口预置为方式1或方式2时，再向控制寄存器写入“C口置0/置1控制字”。这一操作的主要目的是使相应端口的中断允许触发器置0，从而禁止中断，或者使相应端口的中断允许触发器置1，从而允许端口提出中断请求；

注意：“C口置0/置1控制字”虽然是对C口进行操作，但是该控制器是命令字，所以要写入控制寄存器，而不是写入C口控制寄存器。

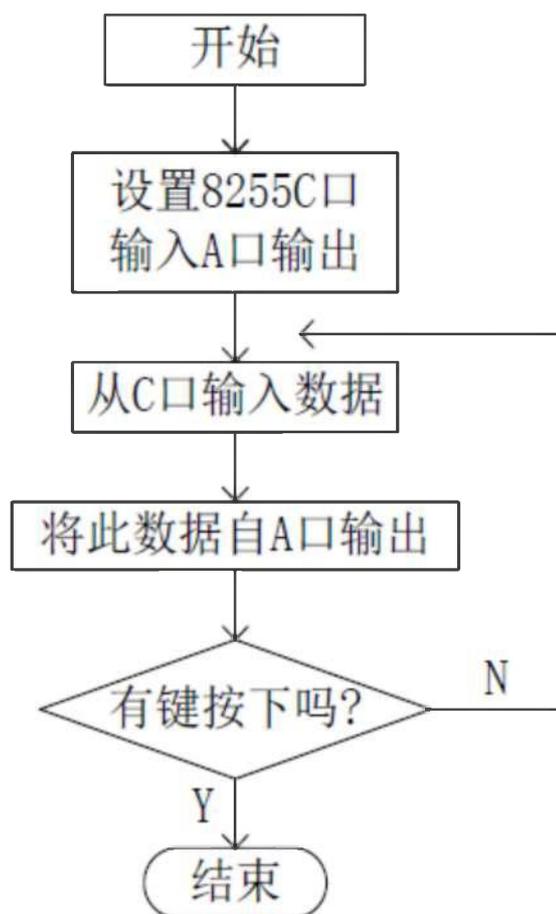
- ③ 向8255数据寄存器写入“数据”或从8255数据寄存器读出“数据”。

三、硬件接线图与软件流程图

硬件接线图：



软件流程图



四、源代码

```
CODE SEGMENT
    ASSUME CS:CODE
START:MOV AL,89H
    MOV DX,28BH
    OUT DX,AL
LOOP1:MOV DX,28AH
    IN AL,DX
    MOV DX,288H
    OUT DX,AL
    MOV AH,1
    INT 16H
```

```
JZ LOOP1
CODE ENDS
END START
```

五、实验结果

将一个拨码开关推上去，其对应的LED灯亮。

六、实验收获与心得体会

这个实验比较简单，代码也很简短，但通过这个实验对8255有了初步的了解，对于接下来的实验有很大帮助。

实验四 七段数码管

一、实验目的

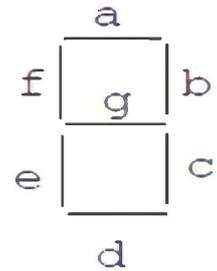
- 1、掌握数码管显示字的原理。

二、实验内容及原理

1、静态显示：将 8255 的 A口 PA0 ~PA7 分别与七段数码管的驱动输入 分别与七段数码管的驱动输入 端 a~dp 相连，位码驱动输入端 S0 、 S1 、 S2 、 S3 接 PC0 、 PC1 、 PC2 、 PC3 ，编 程在数码管显示自己的学号后四位。（或编上循环“ 程在数码管显示自己的学号后四位。（或编上循环“ 程在数码管显示自己的学号后四位。（或编上循环“ 00 -99 ”， 位 码驱动输入端 S0 、 S1 接 PC0 、 PC1 ； S2 、 S3 接地。） 接地。）

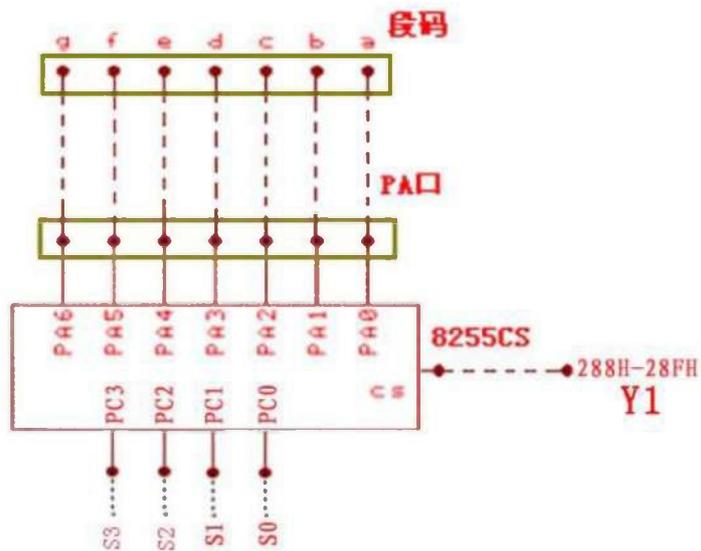
2、实验台上的七段数码管为共阴型，段码采用同相驱动，输入端加高电平，选中的数码管亮，位码加反相驱动器，位码输入端高电平选中。七段数码管的字型代表码如下：

显示字形	g	e	f	d	c	b	a	段码
0	0	1	1	1	1	1	1	3fh
1	0	0	0	0	1	1	0	06h
2	1	0	1	1	0	1	1	5bh
3	1	0	0	1	1	1	1	4fh
4	1	1	0	0	1	1	0	66h
5	1	1	0	1	1	0	1	6dh
6	1	1	1	1	1	0	1	7dh
7	0	0	0	0	1	1	1	07h
8	1	1	1	1	1	1	1	7fh
9	1	1	0	1	1	1	1	6fh



三、硬件接线图与软件流程图

硬件接线图：

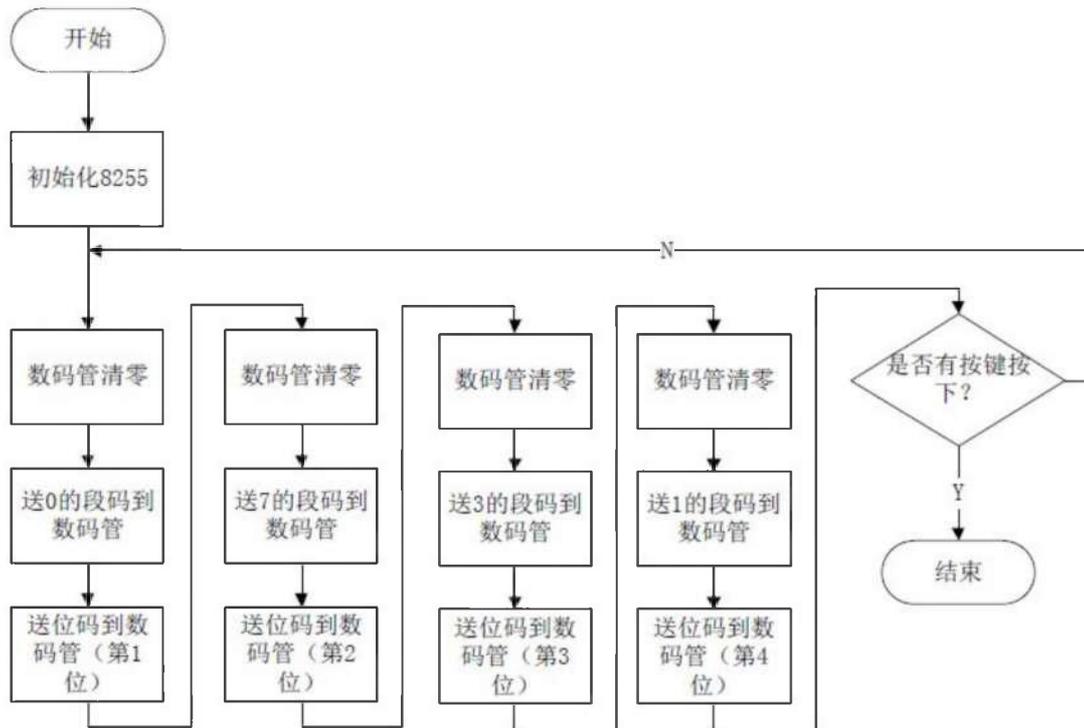


接线：

PA7~PA0/8255 接 dp~a/LED 数码管

PC3~PC0/8255 接 S3~S0/LED 数码管
CS/8255 接 Y1/IO 地址

软件流程图:



四、 源代码

CODE SEGMENT

ASSUME CS:CODE

START: MOV AL,80H

MOV DX,28BH

OUT DX,AL

LOOP1: MOV DX,28AH ;数码管清零

MOV AL,00H

OUT DX,AL

MOV DX,288H ;送出段码

MOV AL,3FH ;0 的段码

```
OUT DX,AL
MOV DX,28AH ;送出位码
MOV AL,08H ;最高位显示
OUT DX,AL
```

```
MOV DX,28AH ;数码管清零
MOV AL,00H
OUT DX,AL
MOV DX,288H ;送出段码
MOV AL,07H ;7 的段码
OUT DX,AL
MOV DX,28AH ;送出位码
MOV AL,04H ;次高位显示
OUT DX,AL
```

```
MOV DX,28 AH ;数码管清零
MOV AL,00H
OUT DX,AL
MOV DX,288H ;送出段码
MOV AL,4FH ;3 的段码
OUT DX,AL
MOV DX,28AH ;送出位码
MOV AL,02H ;第 3位显示
OUT DX,AL
```

```
MOV DX,28AH ;数码管清零
MOV AL,00H
OUT DX,AL
MOV DX,288H ;送出段码
MOV AL,06H ;1的段码
T DX,AL
MOV DX,28AH ;送出位码
```

```
MOV AL,01H ;第 4位显示  
OUT DX,AL
```

```
MOV AH,1 ;调用 DOS 功能，监视键盘是否有输入  
INT 16H  
JZ LOOP1 ;键盘有输入跳出循环，否则继续显示  
MOV AH,4CH ;返回 DOS  
INT 21H
```

```
CODE ENDS
```

```
END START
```

五、实验结果

数码管上静态显示我的学号后四位 0731 。

六、实验总结

在本实验中数码管的显示是一位写，当然也可以用查表方法来进行。在本实验中数码管的显示是一位写，当然也可以用查表方法来进行。

七、实验收获与心得体会 、实验收获与心得体会

这是第一次在硬件实验中使用到数码管，的显示需要位和段输 这是第一次在硬件实验中使用到数码管，的显示需要位和段输 送。这次实验也真正地通过 8255 控制外设的动作。懂得了置 控制外设的动作。懂得了置 8255 的工作方式以及 C口的置/复位的方法。

实验八 可编程定时器/计数器（8253/8254）

一、实验目的

1. 掌握 8253 用作定时器的编程原理；

二、实验内容及原理

1、8253 芯片

1) 8253 初始化

使用 8253 前，要进行初始化编程。初始化编程的步骤是：

A、向控制寄存器端口写入字对使用的计数规定其方式等。

B、向使用的计数器端口写入初值。

2) 8253 控制字：

D7	D6	D5	D4	D3	D3	D1	D0
计数器选择		读写方式选择			工作方式选择		数制选择

D7D6=00：使用0号计数器，D7D6=01：使用1号计数器

D7D6=10：使用2号计数器，D7D6=11：无效

D5D4=00：锁存当前计数值

D5D4=01：只写低8位（高8位为0），读出时只读低8位

D5D4=10：只写高8位（低8位为0），读出时只读高8位

D5D4=11：先读/写低8位，后读/写高8位计数值

D3D2D1=000：选择方式0，D3D2D1=001：选择方式1

D3D2D1=X10：选择方式2，D3D2D1=X11：选择方式3

D3D2D1=100：选择方式4，D3D2D1=101：选择方式5

D0=0：计数初值为二进制，D0=1：计数初值为BCD码数

附：

8253 控制寄存器地址 283H 计数器 0地址 280H

计数器 1地址 281H 计数器 2地址 282H

定时器可工作在方式 3下。

综上所述，设置控制字为：00111110B，即选择零号计数器先读 /先写低 8位、再读 /写高 8位，选择方式 3，计数初值为二进制。

2、音乐产生原理

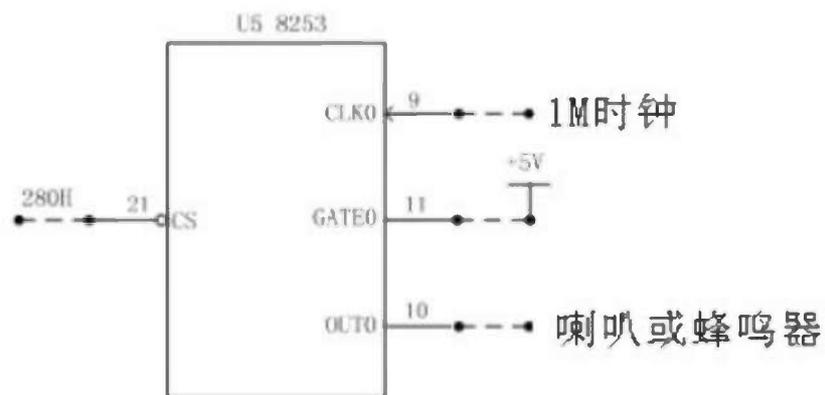
由参考资料中的音符与频率对应表，根据计数器的性质，利用公式，使用MATLAB矩阵运算可以快速计算出七种音符在高中低音时计数器应该对应的初始计数值，如下表所示。

音符频率对照

音符	1 (do)	2 (re)	3 (mi)	4 (fa)	5 (so)	6 (la)	7 (si)	i (do)
频率 (Hz)	256	288	320	341	384	426 . 6	480	512

三、硬件接线图与软件流程图

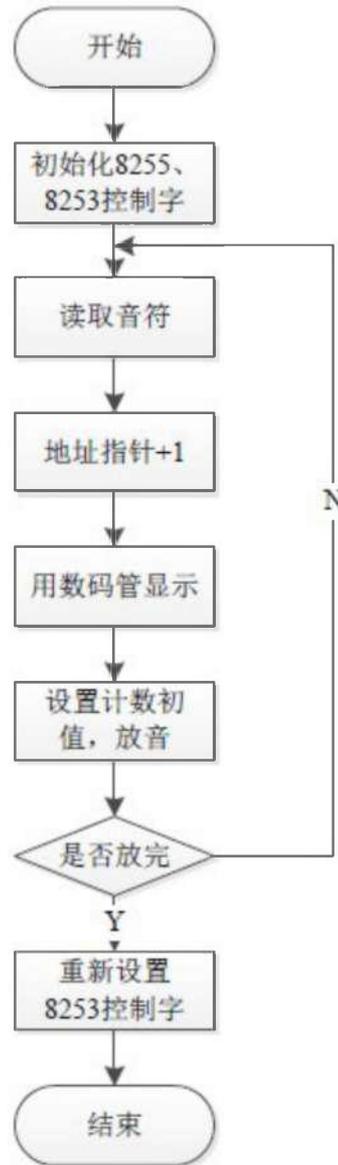
硬件接线图：



接线：

- CS /8253 接 Y0 /IO 地址
- GATE0 /8253 接 +5V
- CLK0 /8253 接 1M 时钟
- OUT0 /8253 接 喇叭

软件流程图：



四、源程序

DATA SEGMENT

BUF DB '334554321123322';欢乐颂乐谱

LTH EQU \$-BUF ;乐谱长度

ENDS

STACK SEGMENT

DB 100 DUP(?)

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:STACK

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/216111053115011004>